Перспективы WorkFlow-систем. Сравнение workflow-языков

Андрей Михеев, Михаил Орлов,

В предыдущих статьях этого цикла (см. PC Week/RE, № 23/2004, с. 26; № 28/2004, с. 21; № 43/2004, с. 36) мы уже рассказывали, что такое системы workflow (WF). Напомним некоторые сведения, которые нам потребуются в данной статье.

WF-системы реализуют процессный подход к управлению предприятием. Для описания автоматизируемых бизнес-процессов в них применяются специальные языки (WF-языки). Задача WF-языка - описать бизнес-процесс формально: задать его возможные состояния, в которых определены соответствующие действия, определить набор внутренних переменных, бизнес-правила, графические элементы форм, связать действия бизнес-процесса с соответствующими внешними приложениями и ролями пользователей и т. д.

В настоящее время не существует единого стандартного WF-языка: между международными коалициями, разрабатывающими WF-стандарты, идет “война спецификаций”. Сейчас насчитывается более десяти несовместимых друг с другом стандартов, относящихся к управлению бизнес-процессами. Наиболее известными являются следующие WF-языки:

- язык XPDL (коалиция WfMC);
- язык BPML (коалиция BPMI);
- язык BPEL4WS (коалиция IBM, Microsoft, BEA, SAP и Siebel).

Нынешние коммерческие системы скорее тяготеют к языку BPEL4WS, а решения OpenSource, как правило, реализуют язык XPDL (и другие стандарты коалиции WfMC). Некоторые системы поддерживают экспорт определений бизнес-процессов при помощи нескольких спецификаций.

В условиях отсутствия единого стандарта группой ученых была предложена методология систематизации и классификации WF-систем и стандартов. Они проанализировали распространенные WF-системы и стандарты, выделили в них типичные элементы, выявили наиболее часто повторяющиеся структуры и назвали их WorkFlow-паттернами.

Краткое описание WF-языка XPDL

В основе языка XPDL лежит математическое понятие -- ориентированный граф. Граф представляет собой набор узлов, частично соединенных переходами. Изменение состояния бизнес-процесса соответствует переходу точки управления из одного узла графа в другой.

Основные элементы языка:

- Activity (узел-действие);

- Transition (переход);
- Participant (участник бизнес-процесса);
- Application (внешнее приложение);
- DataType (тип переменной);
- DataField (переменная).

Как используются элементы языка.

Действия и порядок их выполнения

Граф бизнес-процесса определяется наборами элементов Activity и Transition.

Activity - это основной элемент бизнес-процесса. Элементы Activity соединяются при помощи элементов Transition. Существует три типа элементов Activity:

- Route;
- Implementaion;
- BlockActivity.

Route - узлы, не выполняющие действий, -- используются в целях маршрутизации точек управления.

Inplementation - узлы, с которыми связаны действия в бизнес-процессе. Существует три варианта Inplementation:

- узел взаимодействия с пользователем;
- узел взаимодействия с внешним приложением;
- узел, запускающий подпроцесс.

BlockActivity - узел-контейнер, содержащий в себе не имеющую разветвлений последовательность узлов.

Элемент Transition используется для описания переходов между элементами Activity.

Каждый такой элемент содержит информацию о том, между какими Activtiy и при каких условиях осуществляется переход (переходы бывают условные и безусловные).

Элементы, описывающие данные бизнес-процесса

В начале описания workflow-процесса находятся спецификации типов (тег TypeDeclaration). Для описания данных, относящихся к процессу, и параметров, передаваемых и возвращаемых приложениями, используются элементы DataField и DataType. Кроме того, в XPDL существует понятие ExtendedAtributes - оно дает возможность расширять язык путем ввода дополнительных типов переменных.

Элементы, описывающие исполнителей заданий

Для описания участников бизнес-процесса, т. е. сущностей, которые могут выполнять работу, используется элемент Participant. Существует шесть подтипов элемента Participant:

- Role - соответствует роли участника бизнес-процесса. Для каждой роли должен существовать список людей, которые могут быть “назначены” на эту роль;
- OrganisationUnit - соответствует административному подразделению организации;
- Human - конкретный человек, который будет взаимодействовать с бизнес-процессом при помощи графического интерфейса;
- System - конкретное приложение;
- Resource - некоторый ресурс, который может предложить исполнителя работы;
- RecourceSet - набор ресурсов.

Взаимодействие с внешними приложениями

В языке XPDL задаются спецификации внешних приложений - фактически это описание функций: их названия и параметры (при помощи тега Application). Внутри Activity конкретное приложение указывается в виде параметра тега Tool, и внутри этого тега также производится отображение формальных параметров на фактические.

Структура бизнес-процесса

Упрощенно описание бизнес-процесса в XPDL выглядит следующим образом.

- Определяются переменные бизнес-процесса (и их типы).
- Описываются участники бизнес-процесса (роли и т. д.).
- Задается множество внешних приложений, вызываемых бизнес-процессом (имена функций и типы их параметров).
- Описывается множество всех узлов графа бизнес-процесса, для узла задаются исполнитель, фактические параметры, набор исходящих переходов и т. д.
- Определяются все переходы. Для каждого перехода указывается, какие узлы он связывает, и в случае необходимости условие, при котором по данной связи осуществляется передача управления.

Переход может соединять любые два узла, то есть бизнес-процессу может соответствовать граф любой сложности и топологии. В частности, в графе бизнес-процесса допустимы циклы (WF-паттерн “произвольный цикл”). Поддержка “произвольных циклов” в WF-языке аналогична допустимости использования оператора “goto” в обычных языках программирования.

Однако в силу того, что переход управления осуществляется только по ребрам графа, в XPDL нельзя реализовать WF-паттерн “отложенный выбор”.

Выполнение бизнес-процесса

Технология работы с определениями и экземплярами бизнес-процессов, записанных на языке XPDL, определяется другими спецификациями коалиций WfMC и OMG:

- OMG. Workflow Management Facility Specification;
- WfMC. WAPI (Workflow Application Programming Interface).

В этих спецификациях описывается общая архитектура workflow-системы -- интерфейсы взаимодействия различных компонентов системы друг с другом. В частности, спецификации определяют интерфейсы взаимодействия клиентского приложения и внешней системы с ядром workflow-системы, в которое загружен бизнес-процесс. Эти интерфейсы содержат такие команды, как “запустить процесс”, “посмотреть состояние процесса” и т. д. Основными команды, относящимися к работе с экземпляром бизнес-процесса, являются:

- “Сгенерировать список текущих заданий”;
- “Сообщить ядру, что данное задание выполнено”.

То есть предполагается, что ядро системы, реализующей XPDL, полностью пассивно - оно только отвечает на действия взаимодействующих с ним субъектов.

Краткое описание WF-языка BPML

BPML - язык, основанный на XML и ориентированный на Web-сервисы. В отличие от XPDL, он принадлежит к так называемым структурно-ориентированным языкам: бизнес-процесс в BPML соответствует не математическому графу, а иерархическому набору вложенных и последовательных тегов.

Назовем основные элементы, при помощи которых определяется workflow-процесс в BPML:

- Activity (узел - действие);
- Context (контекст);
- Property (свойство);
- Signal (сигнал);
- Exception (исключение).

Элементы, определяющие выполняемые действия и порядок их выполнения

Activity - основной элемент бизнес-процесса. Элементы Activity могут соединяться последовательно или вкладываться один в другой. Соответственно Activity могут быть простыми (не содержащими других Activity) или сложными. В описании языка указано, что бизнес-процесс является специальным типом сложной Activity. Всего существует 17 типов простых Activity. Основной тип простой Activity называется Action. Когда управление бизнес-процессом попадает в Activity этого типа, происходит вызов описанных там Web-сервисов. Есть типы Activity, соответствующие ветвлению процесса (ветвление относится только к содержащимся внутри них Activities), - это Switch и All Activities. Также существуют типы Activities, которые запускают дочерние процессы (как с ожиданием их окончания, так и без), организуют задержки выполнения процесса и т. д. Кроме того, есть несколько типов Activities, реализующих разного вида циклы. Для синхронизации точек управления, находящихся в Activities одного уровня вложенности, в языке используются сигналы (Signals).

В BPML также введены понятия “исключение” (Exception) и “компенсация” (Compensation). “Исключение” соответствует возникновению нештатной ситуации, когда оказывается, что выполнять некоторый участок бизнес-процесса уже не требуется. (Например, во время выполнения бизнес-процесса оформления туристической поездки клиент позвонил в туристическую компанию и сообщил, что он отказывается от поездки.) “Компенсация” соответствует необходимым действиям по корректному завершению ситуации, возникшей в связи с Exception. (Если в вышеприведенном примере для клиента были забронированы билеты на самолет и номер в гостинице, то задачей Compensation будет отменить бронирование.)

Элементы, описывающие данные бизнес-процесса

На языке BPML данные описываются в рамках контекста (Context). Контекст содержит относящиеся к процессу переменные, локальные определения процессов, сигналов и т. д., служит для передачи информации между узлами и синхронизации.

Переменные определяются тегом Property. Они могут быть локальными или глобальными по отношению к данному контексту.

Исполнители в спецификации практически не описываются - все это “перенесено” на технологию Web-сервисов. Не определен в языке BPML и синтаксис конструкций для взаимодействия бизнес-процесса с внешними приложениями: здесь тоже используется технология Web-сервисов.

Сравнение BPML с XPDL

Язык BPML существенно “легче” языка XPDL.

- В нем не надо описывать внешние приложения (Applications в XPDL) - эти

функции “перекладываются” на технологию Web-сервисов.

- Не надо определять и “присоединять к Activities” переходы. Архитектура связей определяется вложенностью тегов, соответствующих элементам Activity, - в BPML нет понятия Transition.
- Не надо в рамках языка специально определять описание участника бизнес-процесса. Все участники - это Web-сервисы; следовательно, соответствующие описания отвечают спецификации Web-сервисов.

Кроме того, для работы с бизнес-процессами, написанными на XPDL, требуются дополнительные спецификации. В частности, они указывают, каким образом можно “сообщить” определенной Activity, что она выполнена и управление может двигаться дальше, и т. д. В соответствии с идеологией языка XPDL среда, в которой выполняется бизнес-процесс, не является активной; активными должны быть внешние участники, а среда выполнения процесса только реагирует на их действия.

“Идеология” языка BPML скорее противоположна - в ней бизнес-процесс может быть активным и давать задания участникам-исполнителям. Исполнители, являясь Web-сервисами, могут “ничего не знать” о бизнес-процессе, в котором они участвуют.

Однако отказ от понятия Transition и замена его вложенностью тегов приводят к тому, что граф, соответствующий бизнес-процессу, в BPML не может быть графом произвольной структуры, например, он не может содержать сложные циклы. Вследствие этого в BPML невозможно реализовать WF-паттерн “Произвольный цикл”. Для того чтобы выполнять повторяющиеся последовательности шагов, в BPML введено несколько типов Activity-циклов, однако в этом случае циклически повторяться может только содержащаяся внутри данной Activity последовательность узлов.

Отсутствию произвольных циклов в бизнес-процессе можно поставить в соответствие аналогию программирования “без goto”.

Замечание. В BPML параллельно выполняющиеся ветви процесса могут дополнительно обмениваться сигналами, производя, таким образом, синхронизацию. Трудно сказать, облегчает это понимание бизнес-процесса или, наоборот, затрудняет.

В силу того, что язык BPML основан на тегах, в нем легко организовать генерацию и обработку исключений. А не будучи явно основанным на теории графов, он позволяет реализовать WF-паттерн “отложенный выбор”. В BPML поддерживаются такие концепции, как транзакции и расписания. В нем можно устанавливать задержки и deadlines. Недавно и в XPDL появились понятия TimeOuts и Exceptions, однако функциональность их заметно ниже соответствующих конструкций BPML.

Краткое описание WF-языка BPEL4WS

Язык BPEL4WS также ориентирован на Web-сервисы. Во многом он похож на BPML, однако существенно сложнее его. Появился BPEL4WS путем слияния WF-языков WSFL и XLANG. Эти языки основаны на разных моделях: WSFL базируется на теории графов, XLANG - на иерархии тегов XML.

BPEL4WS унаследовал конструкции обоих языков. Например, он допускает реализацию некоторых WF-паттернов в двух вариантах: в стиле WSFL и в стиле XLANG. В некоторых случаях допустим и смешанный стиль. Это делает BPEL4WS трудным для изучения. Несмотря на то что BPEL4WS является наследником языков различной природы, его можно отнести к классу структурно-ориентированных: унаследованные “граф-ориентированные” конструкции реально соответствуют некоторым ограничениям на порядок выполнения Activities внутри параллельного блока.

BPEL4WS определяет два вида процессов - абстрактный и исполняемый. Абстрактный процесс определяет протокол обмена сообщениями между различными участниками, не “открывая” алгоритмы их “внутреннего” поведения. В отличие от абстрактного исполняемый процесс содержит в себе алгоритмы, определяющие порядок выполнения Activities, назначение исполнителей, обмен сообщениями, правила обработки исключений и т. д.

Activities в BPEL4WS делятся на примитивные и структурные.

Примитивные Activities:

- Receive - ожидает сообщения внешнего источника;
- Reply - отвечает внешнему источнику;
- Invoke - “запускает” операцию какого-либо Web-сервиса;
- Wait - ждет в течение определенного периода времени;
- Assign - копирует значение одной переменной в другую;
- Throw - отбрасывает исключение в случае ошибки;
- Terminate - принудительно завершает выполнение службы;
- Empty - не выполняет никаких действий.

Структурные Activities:

- Sequence - соответствует последовательному выполнению Activities, содержащихся внутри этого элемента.
- Switch - условная передача управления (соответствует оператору Switch языков программирования C++, Java и т. д.);
- While - организует цикл типа “While”;
- Pick - запускает обработку событий и исключительных ситуаций;
- Flow - соответствует параллельному выполнению Activities, содержащихся внутри этого элемента.
- Scope - группирует узлы для программы -- обработчика ошибок.

Кроме того, в языке присутствует понятие “связь” (link). Эта конструкция унаследована из граф-ориентированного “предка” BPEL4WS. Как правило, применяется она к Activities, находящимся внутри параллельного блока, и накладывает ограничения на порядок их выполнения. К конструкции языка, использующей link, может быть применено, например, такое ограничение: Activities, соединенные при помощи этого элемента, не могут образовывать циклов.

Переменные описываются при помощи тега “variables”. Для задания исполнителя используется тег “partnerLink”, в языке активно используется технология Web-сервисов.

Общение с “внешним миром” в BPEL4WS определяется технологией Web-сервисов. В некоторые теги добавлен параметр “variable”.

Идеологически языки BPEL4WS и BPML очень похожи. Нам кажется, что BPML проще и удобнее, чем BPEL4WS, однако за BPEL4WS стоят такие корпорации-гиганты, как IBM, Microsoft, BEA, SAP и Siebel, и вполне возможно, что благодаря “маркетинговой мощи” этих компаний язык BPML будет полностью вытеснен языком BPEL4WS.

jPDL - “нестандартный” язык, ориентированный на поддержку WF-паттернов.

В последнее время появился еще один подход к построению языков описания бизнес-процессов. Он вызван к жизни следующими двумя обстоятельствами:

в настоящее время не существует единого общепринятого WF-языка - между коалициями идет “война стандартов”, и еще непонятно, какой из них окажется победителем;

многие стандарты, относящиеся к области управления бизнес-процессами, выглядят неоправданно сложными.

С их учетом при разработке WF-систем имеет смысл не реализовывать полностью какую-либо из приведенных выше спецификаций, а использовать упрощенные языки описания бизнес-процессов, поддерживающие основные WF-паттерны. В этом случае сами WF-системы будут значительно проще и надежнее, а поддержка наиболее расропространенных WF-паттернов гарантирует некоторый базовый уровень функциональности системы.

Примером такого языка является jPDL (проект JBOSS jBPM).

jPDL производит впечатление языка в какой-то степени “промежуточного” между XPDL и BPML, однако он гораздо ближе к XPDL, чем к BPML. В случаях обычных переходов между узлами и выбора одного направления из нескольких возможных это аналог концепции WfMC, а при параллельном расщеплении потоки описываются внутри специального тега - “concurrent block”, что скорее соответствует идеологии BPML.

jPDL несомненно относится к классу граф-ориентированых языков и разработан в духе идеологии WfMC; фактически он соответствует упрощенному варианту спецификаций этой коалиции.

Основным упрощением jPDL является то, что во многих случаях вместо сложных конструкций XPDL этот язык напрямую использует классы и конструкции языка программирования Java.

Основные элементы языка jPDL:

- State (узел-действие);
- Process-state (узел-подпроцесс);
- Decision (исключающий выбор);
- Fork (параллельное расщепление);
- Join (синхронизация);

- Swimlane (роль-дорожка);
- Variable (переменная);
- Transition (переход).

В соответствии с правилами языка jPDL бизнес-процесс определяется файлом-архивом, содержащим несколько XML-описаний, в которых задаются узлы графа бизнес-процесса, переходы между ними, роли-дорожки участников бизнес-процесса и его переменные. Также архив содержит описания HTML-форм, используемых в соответствующих узлах бизнес-процесса, Java-классы, специфичные именно для данного бизнес-процесса и подгружаемые в ядро системы при вызове бизнес-процесса, а также дополнительную информацию, например визуальное представление графа бизнес-процесса, на котором пользователю будут показано текущее положение точек управления.

Переменные описываются при помощи тега “variable”. Для определения “фактического” типа переменной разрешается указывать Java-классы. Исполнители указываются при помощи тегов “swimlane” и “assignment”, для выделения инициализатора можно в тег ”swimlane” вставлять соответствующую ссылку на Java-класс. Приложения в jPDL ничем не отличаются от обычных пользователей. Дополнительно в языке существует механизм обращения к специальным Java-классам, которые можно разместить в архиве бизнес-процесса (тег “action”).

Оказалось, что отказ от следования стандартам международных коалиций и ориентация на язык программирования Java принесли jPDL много преимуществ:

- язык оказался простым, но достаточно мощным;
- ядро OpenSource workflow-системы JBOSS jBPM, интерпретирующее jPDL, также является простым и понятным для большого количества Java-программистов, что обусловило рост его популярности и включение его в линейку продуктов JBOSS;

Ориентация на Java способствует “разделению труда” между программистом и менеджером при разработке бизнес-процессов -- программист реализует некоторые часто используемые компоненты, а менеджер проектирует бизнес-процесс, используя разработанные программистом компоненты, но не вдаваясь в механизм их функционирования.

Иными словами, опыт проекта JBOSS jBPM позволяет поверить в то, что сегодня решения, ориентированные не на поддержку стандартов международных коалиций, а на удовлетворение некоторого класса требований к функциональности (например, совместимости с основными WF-паттернами) могут оказаться вполне успешными.

В данной статье мы привели лишь краткое описание наиболее распространенных WF-языков. Более подробно сравнение языков описано в приложении к статье (pdf, 450 КБ), причем для каждого языка реализованы наиболее характерные WF-паттерны. Кроме того, там детально прокомментированы различия конструкций языков.

В силу несовершенства всех существующих WF-стандартов как для разработчиков софта, так и для организаций, выбирающих WF-систему, весьма опасна “жесткая” привязка к какому-то одному WF-стандарту. Велика вероятность, что в будущем этот стандарт будет кардинально переработан, может быть, даже все современные WF-языки будут вытеснены новым, более удобным и принципиально другим. Таким образом, имеет смысл выбирать

гибкую WF-систему, допускающую импорт-экспорт в различные языки и настройку на новые WF-стандарты, которые еще не существуют.

С авторами статьи, сотрудниками консалтинговой группы “Руна”, можно связаться по адресу: wf@runa.ru.

Литература и ссылки.

1. *W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, A.P. Barros*. Workflow Patterns. 2002.
(http://tmitwww.tm.tue.nl/research/patterns/download/wfs-pat-2002.pdf)

2. *Van der Aalst.* Patterns and XPDL: A critical evaluation of the XML process definition language.
(http://tmitwww.tm.tue.nl/research/patterns/download/ce-xpdl.pdf)

3. *Van der Aalst, M. Dumas, A.H.M. ter Hofstede, P. Wohed*. Pattern based analysis of BPML.
(http://xml.coverpages.org/Aalst-BPML.pdf)

4. *P. Wohed , Van der Aalst, M. Dumas, A.H.M. ter Hofstede.* Pattern based analysis of BPEL4WS.
(http://xml.coverpages.org/AalstBPEL4WS.pdf)

5. *R Shapiro*. A comparison of XPDL, BPML and BPEL4WS.
(http://xml.coverpages.org/Shapiro-XPDL.pdf)

6. Коалиция WfMC, язык XPDL.
(http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf)

7. Коалиция BPMI, язык BPML.
(http://www.bpmi.org/bpml-spec.esp)

8. IBM, Microsoft и BEA, язык BPEL4WS.
(http://www-106.ibm.com/developerworks/library/ws-bpel/)

9. jPDL Reference Manual.
(http://www.jbpm.org/jpdl.html)

10. OMG. Workflow Management Facility Specification.
(http:// www.omg.org/docs/formal/00-05-02.pdf)

11. WfMC. WAPI (Workflow Application Programming Interface).
(http://www.wfmc.org/standards/docs/TC-1009_20e_1997.pdf)

12. *А. Михеев, М. Орлов.* Перспективы workflow-систем// PCWeek, № 43/2004, с. 36
(http://kis.pcweek.ru/Year2004/N43/CP1251/CorporationSystems/chapt2.htm)